

INTRODUÇÃO AOS SISTEMAS DE COMPUTAÇÃO

Prezado(a) aluno(a),

Agora que você já tem uma visão geral de Sistemas Operacionais, apresentaremos neste capítulo uma introdução aos Sistemas de Computação, que se dividem em software, hardware e elemento humano.

O item Software é essencial para a contextualização do sistema operacional. Dê atenção especial aos conceitos de compiladores, interpretadores, linkers e loaders, pois eles serão necessários para o entendimento de capítulos futuros.

Os conceitos fundamentais de hardware você aprendeu na disciplina: Organização Estruturada de Computadores, por isso, trataremos o assunto de forma mais breve.

Quanto ao elemento humano, lembre-se sempre de que ele é o elemento mais importante de todo sistema de computação.

Bom estudo!

que risus ac
ne velit at tellus.
massa porttitor
sectetur magna.

Fala Professor

2.1. Introdução

Um Sistema Operacional é um componente importante de praticamente todo sistema de computação, que se constitui por: Software, Hardware e Elemento Humano. Espera-se de um sistema de computação que ele execute programas de usuários e permita a solução de problemas. Para alcançar esses dois objetivos, o sistema operacional é colocado entre o hardware de computador e os programas dos usuários. Os três componentes de um sistema de computação devem trabalhar de forma conjunta. Veja, na Figura 2-6, uma breve descrição de cada um deles.

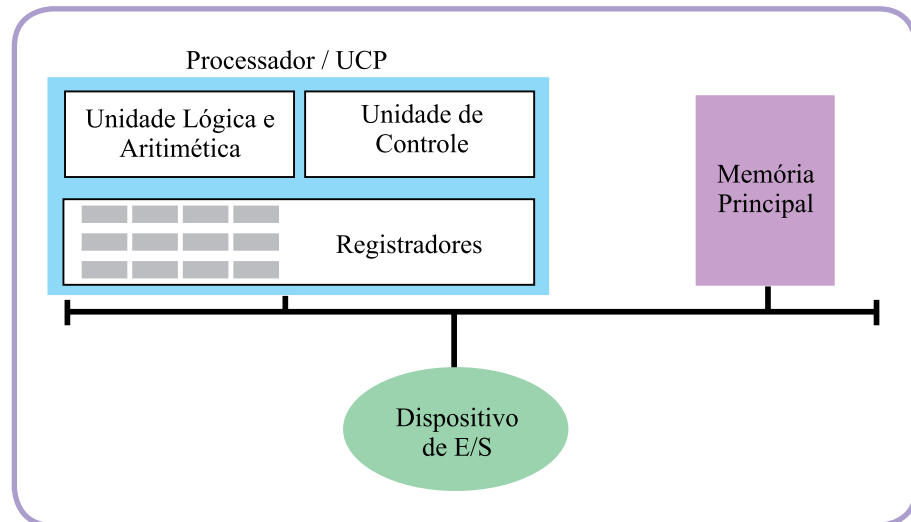


Figura 2-6: Sistema de Computação

Fonte: [1] – Machado e Maia, 2004. Adaptação.

2.2. Software

Para que o hardware tenha utilidade prática, deve existir um conjunto de programas, utilizados como interface entre as necessidades do usuário e as capacidades do hardware. O software de um sistema de computação pode ser dividido em: Software Básico, Utilitário e Aplicativo.

2.2.1. Software básico

Conceitos



Software básico é um software para controlar ou coordenar o uso do hardware entre vários programas aplicativos e usuários [1].

Para controlar o hardware, os softwares básicos geralmente necessitam de se comunicar diretamente com o hardware para gerenciá-lo (exemplo: drivers de dispositivos). O sistema operacional é composto por um conjunto de softwares básicos.

2.2.2. Utilitários

Utilitários são programas para auxiliar a utilização de um Sistema Operacional. A interface gráfica de usuário (GUI – *Graphic User Interface*) e a interface linha de comando são exemplos de utilitários [1].

que risus ac
ne velit at tellus.
massa porttitor
sectetur magna.

Fala Professor

Os utilitários, em geral, são instalados juntamente com o sistema operacional. O termo utilitário também serve para referenciar softwares que estão relacionados mais diretamente com o sistema operacional, tais como: compiladores, *linkers* (ligadores), *loaders* (carregadores).

a) Tradutor

Nos primeiros sistemas computacionais, os programas eram desenvolvidos em linguagem de máquina e carregados diretamente na memória principal para execução. Isso tornava a programação bastante complexa, pois o programador deveria conhecer a arquitetura da máquina e programar em painéis, por meio de fios.

O surgimento das primeiras linguagens de montagem ou *assembly* e das linguagens de alto nível facilitou a tarefa de programação, pois o programador deixou de se preocupar com muitos aspectos específicos do hardware para desenvolver programas, facilitando também a manutenção e a documentação.

Apesar das vantagens proporcionadas pelas linguagens de montagem e de alto nível, os programas escritos em tais linguagens não estão prontos para serem executados pelo processador (programas-fonte). Para isso, eles têm que passar por uma etapa de conversão, quando toda representação simbólica das instruções é traduzida para o código de máquina.

O módulo gerado pelo tradutor é denominado módulo objeto e, apesar de estar em código de máquina, na maioria das vezes não pode ser executado. Isso ocorre em função de um programa poder chamar sub-rotinas externas, e, nesse caso, o tradutor não tem como associar o programa principal às sub-rotinas chamadas. Essa função é realizada por outro utilitário denominado “linker”, e será apresentada adiante.

Dependendo do tipo do programa-fonte serão utilizados tradutores distintos: montador e compilador.

Conceitos



Montador (*assembler*) é o utilitário responsável por traduzir um programa-fonte em linguagem de montagem num programa objeto não executável (módulo objeto) [1].

A linguagem de montagem é particular para cada processador, assim como a linguagem de máquina, o que impede que programas escritos em *assembly* possam ser portados em máquinas diferentes.

Conceitos



Compilador é o utilitário responsável por traduzir um programa-fonte escrito em linguagem de alto nível num programa em linguagem de máquina [1].

Os programadores de alto nível devem se preocupar apenas com o desenvolvimento de suas aplicações, não tendo que conhecer detalhes sobre a arquitetura do processador. Assim, programas-fonte em linguagem de alto nível podem ser portados entre computadores de diversos fabricantes, desde que exista padrão para a sintaxe da linguagem. Isso permite o desenvolvimento de aplicações independentes do equipamento.

b) Interpretador

O interpretador é considerado um tradutor que não gera módulo-objeto. A partir de um programa-fonte escrito em linguagem de alto nível, o interpretador, durante a execução do programa, traduz cada instrução e a executa imediatamente.

A maior desvantagem dos interpretadores é o tempo gasto na tradução das instruções de um programa toda vez que este for executado, já que não existe a geração de um código executável. A vantagem é a portabilidade, pois um programa fonte pode ser executado em qualquer máquina que possua o seu respectivo interpretador.

c) Linker (Ligador)

O *Linker* ou Ligador é o utilitário responsável por gerar, a partir de um ou mais módulos-objeto, um programa executável. Suas funções básicas são resolver todas as referências simbólicas existentes entre os módulos e reservar memória para a execução do programa. Para resolver todas as

referências a símbolos, o *linker* pode pesquisar em bibliotecas do sistema ou do próprio usuário. Bibliotecas são arquivos que contêm diversos módulos-objeto e/ou definições de símbolos.

Bibliotecas são arquivos que contêm diversos módulos-objeto e/ou definições de símbolos.



Conceitos

Outra função importante do linker é a relocação, que determina a região de memória na qual o programa será carregado para execução. Nos primeiros sistemas operacionais, a relocação era realizada somente uma vez, na etapa de *linkedição*. Todos os endereços simbólicos do programa eram traduzidos para endereços físicos, e o programa executável gerado podia ser carregado a partir de uma posição prefixada na memória (código absoluto). Nesse tipo de relocação, um programa poderia ser carregado somente em uma única posição pré-determinada na memória.

Em sistemas multiprogramáveis, a utilização de código absoluto torna-se inviável, pois nada garante que uma determinada região de memória esteja disponível para o programa ser carregado. A solução é permitir que um programa possa ser executado em qualquer região disponível da memória, durante a sua carga (código relocável). Esse tipo de relocação não é realizado pelo *linker*, e sim por outro utilitário, chamado *Loader*, responsável por carregar os programas na memória.

d) Loader (Carregador)

Loader ou carregador é o utilitário responsável por carregar um programa da memória secundária para a memória principal, para que possa ser executado.



Conceitos

Quando o loader carrega um programa para a memória principal, ele aloca uma área de código, uma área de dados e uma área de pilha. A área de código armazena o programa executável, a área de dados armazena as variáveis e constantes utilizadas no programa, e a área de pilha armazena os endereços de retornos das funções ou procedimentos chamados durante a execução do programa. O funcionamento do carregador de-

pende do código gerado pelo *linker*, pois, de acordo com este, o *loader* pode ser classificado como do tipo absoluto ou relocável.

Se o código executável for do tipo absoluto, o *loader* só necessita conhecer o endereço de memória inicial e o tamanho do módulo para realizar o carregamento. Então, o *loader* transfere o programa da memória secundária para a memória principal e inicia sua execução (*loader* absoluto). No caso de código relocável, o programa pode ser carregado em qualquer posição de memória, e o loader é responsável pela relocação no momento do carregamento (loader relocável).

2.2.3. Aplicativos

Conceitos



Os **Aplicativos** são programas de usuários instalados para solução de problemas (exemplo: editores de texto, planilhas, etc.).

Na literatura de sistemas operacionais, não há uma definição exata para diferenciar utilitários de aplicativos. Por isso, os conceitos utilitários e aplicativos podem se confundir algumas vezes. Mas, geralmente, o termo aplicação ou aplicativo é utilizado para referenciar programas que estão mais próximos do usuário.

2.3. Hardware

O Hardware de um sistema de computação possui três componentes principais: CPU (Unidade Central de Processamento), Memória e Periféricos. Esses componentes são interligados por meio de um barramento, por onde as informações são transportadas entre eles. Geralmente, em um sistema de computação, o processamento de dados é realizado na seguinte sequência: entrada => processamento => saída. A entrada e a saída são realizadas pelos periféricos, ao passo que o processamento em si, é feito na CPU, com o auxílio da memória.

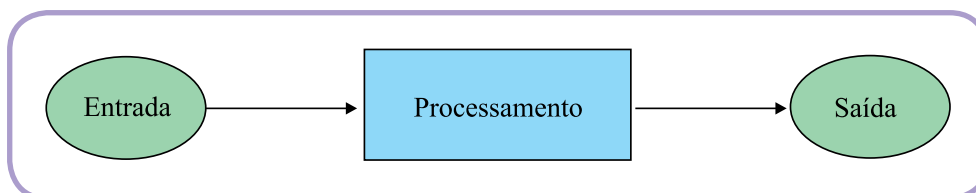


Figura 2-7: Processamento de dados

A CPU possui internamente as seguintes unidades funcionais: ULA (Unidade Lógica-Aritmética), UC (Unidade de Controle) e registradores. A ULA é responsável pelas operações lógicas e aritméticas. A UC envia sinais de controle às unidades dentro da CPU (exemplo: as operações que a ULA deve realizar). Os registradores armazenam informações temporárias dentro da CPU (exemplo: operandos da ULA, instruções para a UC, informações de Status do processador, etc). A principal característica dos registradores é que eles são bastante rápidos (da mesma velocidade da CPU) no acesso às informações por eles armazenadas.

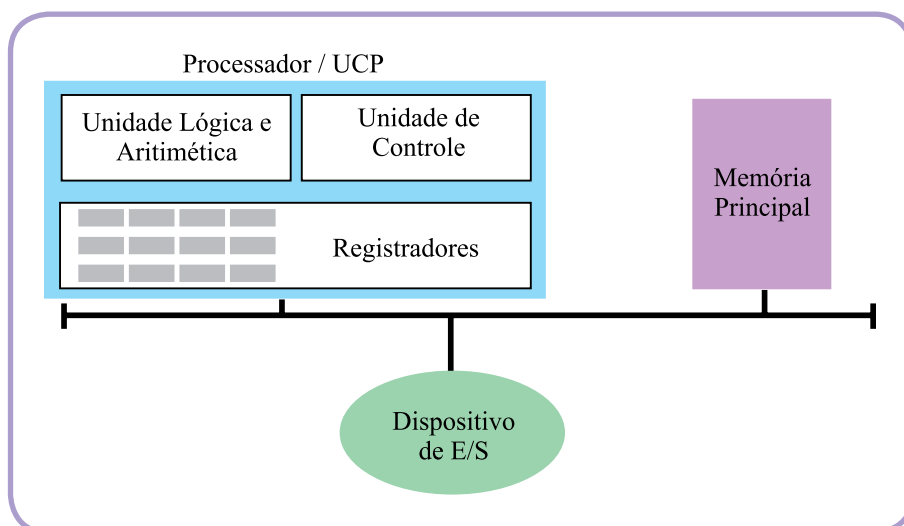


Figura 2-8: Processador, memória e dispositivos de E/S
 Fonte: [1] – Machado e Maia, 2004. Adaptação.

2.3.1. Memória

A memória é o local onde são armazenados dados e instruções. A memória pode ser dividida em: memória principal e memória secundária.

A **memória principal** armazena as informações utilizadas pelos programas que estão sendo executados na CPU. Ela é a única que pode ser acessada diretamente pela CPU, que lê continuamente as instruções armazenadas e as executa. Além disso, a CPU pode armazenar, na memória principal, os dados resultantes de operações efetuadas.



Conceitos

Em geral, a memória principal é volátil (perde o conteúdo ao ser desligada da energia), e, por isso, conta com o auxílio da memória secundária (não-volátil), para armazenar informações, mesmo que o sistema seja desligado.

Conceitos



A **memória secundária** é responsável por armazenar os dados e programas dos usuários de forma não-volátil. A principal diferença entre ela e a memória principal é que a memória secundária não pode ser acessada diretamente pela CPU.

A memória secundária é composta por todos os dispositivos de armazenamento: discos-rígidos, pen drives, CD's, DVD's, disquetes, fitas, etc. A grande vantagem da memória secundária é sua enorme capacidade, se comparada com a memória principal, e sua desvantagem é o tempo de acesso, ou seja, ela é bem mais lenta que a memória principal.

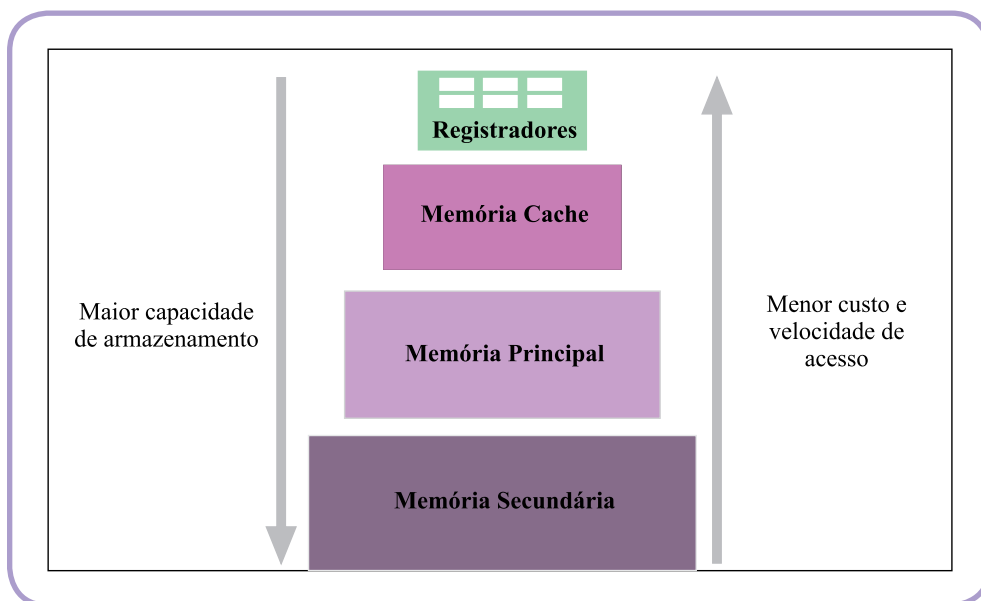


Figura 2-9: Hierarquia de memória

Fonte: [1] – Machado e Maia, 2004. Adaptação.

Visto que a memória principal é volátil, sempre que o sistema for ligado, é necessário carregar o sistema operacional da memória secundária para a memória principal. Esse processo é chamado de **boot** ou ativação do sistema. Em seguida, os programas solicitados pelo usuário também são carregados na memória principal, à medida que são solicitados.

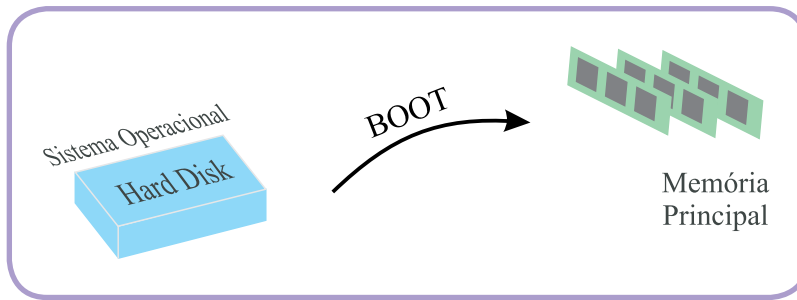


Figura 2-10: Ativação do sistema
Fonte: [1] – Machado e Maia, 2004. Adaptação.

2.3.2. Periféricos

Periféricos são dispositivos de hardware externos à CPU, utilizados para a realização de entrada e saída.



Conceitos

Os Periféricos podem ser classificados em periféricos de: Entrada, Saída e Entrada/Saída. Entre os periféricos de entrada, destacam-se: teclado, mouse, scanner, webcam, microfone, etc.; entre os de Saída destacam-se: monitor, impressora, caixas de som, etc.; e, por fim, os periféricos de Entrada/Saída (E/S) são aqueles que podem realizar entrada e saída. Alguns deles, no entanto, além de realizar E/S também armazenam dados e, por isso, são chamados de periféricos de armazenamento, utilizados para compor a memória secundária: discos rígidos, pendrives, CDs, DVDs, disquetes, fitas. Entre os que realizam estritamente E/S estão: interface de rede, Fax/modem, monitor touch-screen.

2.4. Elemento humano

O Elemento Humano, também chamado de *Peopleware*, compreende todas as pessoas envolvidas em um sistema de computação: usuários finais (que utilizam somente aplicativos), programadores, administradores de rede, analistas de sistemas, desenvolvedores de sistemas, administradores de banco de dados, engenheiros de software, engenheiros de hardware, etc.

Fala Professor

...que risus a
...e velit at tellus.
...massa porttitor
...sectetur magna.

Apesar de nossa disciplina ser voltada para software básico, reforçamos que o elemento humano é o mais importante de todo o sistema de computação. Atitudes como: trabalhar em equipe, ser cordial com os colegas, preocupar-se com o usuário e ser pró-ativo, são essenciais para se alcançar o sucesso na integração do elemento humano em um sistema de computação.

Indicações



[1] MACHADO, F.B. e MAIA, L.P. **Arquitetura de Sistemas Operacionais**. 4.ed. LTC, 2007.

[2] SILBERSCHATZ, A., GALVIN, P.B., GAGNE, G. **Fundamentos de Sistemas Operacionais**. 6.ed. LTC, 2004.

[3] TANENBAUM, A.S. **Sistemas Operacionais Modernos**. 2.ed. Pearson Brasil, 2007.

[4] OLIVEIRA, R.S., CARISSIMI, A.S., TOSCANI, S.S. **Sistemas Operacionais**. 3.ed. Sagra-Luzzato. 2004.

Atividades



Atividades

1. Quais são as unidades funcionais de um sistema computacional?
2. Defina, com suas palavras, o que é: (i) compilador, (ii) interpretador, (iii) linker e (iv) loader.
3. Suponha que você, utilizando a interface gráfica, clicou duas vezes em um ícone. Imediatamente, uma janela se abre e um programa começa a ser executado. Para que isso ocorresse, qual dos utilitários abaixo foi chamado: (i) compilador, (ii) interpretador, (iii) linker ou (iv) loader? Justifique sua resposta.
4. Classifique os programas em (i) softwares básicos, (ii) utilitários e (iii) aplicativos: (a) editor de texto, (b) driver de vídeo, (c) planilha eletrônica, (d) sistema operacional, (e) interface gráfica.
5. Por que o código-objeto gerado pelo tradutor ainda não pode ser executado?

6. Por que a execução de programas interpretados é mais lenta que a de programas compilados?
7. Quais as funções do linker?
8. Qual a principal função do loader?
9. Quais os componentes de um processador e quais são suas funções?
10. Como a memória principal de um computador é organizada?
11. O que são memórias voláteis e não-voláteis?
12. Pesquise como funciona a técnica de pipelining e como ela melhora o desempenho dos sistemas computacionais.

**Conceitos**

